

APPENDIX A

```

int RunPDet (int *ADC_out)
{
    static int PD_In[2]; /*scaled FIR output*/
    if (!Ctrl.AcqTrk) {
        Zpr.PD_Fave = (int)((abs(PD_In[0] - Zpr.PD_ykFd)+1)/2);
        Zpr.PD_Save = (int)((abs(PD_In[1] - Zpr.PD_ykSd)+1)/2);

        Zpr.PD_AmpEst = (int)(sqrt(pow(Zpr.PD_Save,2) + pow(Zpr.PD_Fave,2)));

        Zpr.PD_ykFd = PD_In[0];
        Zpr.PD_ykSd = PD_In[1];

        PD_In[0] = (ADC_out[0];
        PD_In[1] = (ADC_out[1];

        Zpr.PD_condition[1] = Zpr.PD_condition[0];
        Zpr.PD_condition[0] = (Zpr.PD_ykFd + PD_In[0]) < Zpr.PD_Thrsh_Low;

        Zpr.PD_condition[3] = Zpr.PD_condition[2];
        Zpr.PD_condition[2] = (Zpr.PD_ykSd + PD_In[1]) < Zpr.PD_Thrsh_low;
        Zpr.PD_condition[4] = Zpr.PD_AmpEst > Zpr.PD_Thrsh_High;

        if (Zpr.PD_Counter < Zpr.PD_Qual)
            if (Zpr.PD_condition[0] && Zpr.PD_condition[1] &&
                Zpr.PD_condition[2] && Zpr.PD_condition[3] &&
                Zpr.PD_condition[4])
            {
                Zpr.PD_Counter++;
            }
            else {
                Zpr.PD_Counter = 0;
            }
        }
        else {
            Zpr.PD_Fave = Zpr.PD_Save = 0;
            Zpr.PD_ykFd = PD_In[0] = 0;
            Zpr.PD_ykSd = PD_In[1] = 0;
            Zpr.PD_AmpEst = 0;
            Zpr.PD_condition[1] = Zpr.PD_condition[0] = 0;
            Zpr.PD_condition[3] = Zpr.PD_condition[2] = 0;
            Zpr.PD_condition[4] = 0;
            Zpr.PD_Counter = 0;
        }
    }
    return (Zpr.PD_Counter);
}

```

Legend of the code variables with respect to the patent application:

ADC_out[0] = current even sample Se (90c, FIG. 7) from ADC 50 (FIG. 5)

ADC_out[1] = current odd sample Oe (91c, FIG. 7) from ADC 50

PD_In[0] = first previous even sample Se - 1 (90b, FIG. 7)

PD_In[1] = first previous odd sample Oe - 1 (91b, FIG. 7)

Zpr.PD_ykFd = second previous even sample Se - 2 (90a, FIG. 7)
Zpr.PD_ykSd = second previous odd sample Oe - 2 (91a, FIG. 7)

Zpr.PD_Fave = AE (equation 5)
Zpr.PD_Save = AO (equation 6)

Zpr.PD_AmpEst = Amp (equation 7)

Zpr.PD_ykFd + PD_In[0] = E1, E2 (equations 1-2)
Zpr.PD_ykSd + PD_In[1] = 01, 02 (equations 2-4)

Zpr.PD_Thrsh_Low → Threshold_Low (equations 8-11)
Zpr.PD_Thrsh_High → Threshold_High (equation 12)

TOP SECRET